Patient Summary Widget Documentation

This page provides documentation of widgets that can be added to the patient dashboard:

- Section Configuration Parameters
- Latest Obs For Concept List
- Obs Across Encounters
- Obs Graph
- Data Integrity Violations
- Relationships
- Recent Visit by Encounter Type
- Programs List
- Program Status
- Program History
- Bahmni Appointments Widget (Capability).

There are a number of additional features that would be helpful on the dashboard: Future enhancements.

Section Configuration Parameters

The following parameters are required for each section:

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>required</td>
<td>The title of the section</td>
</tr>
<tr>
<td>provider</td>
<td>optional</td>
<td>A valid moduleId</td>
</tr>
<tr>
<td>fragmentId</td>
<td>optional</td>
<td>A valid fragment</td>
</tr>
<tr>
<td>icon</td>
<td>optional</td>
<td>The icon displayed for the heading</td>
</tr>
<tr>
<td>order</td>
<td>optional</td>
<td>How does this work elsewhere</td>
</tr>
<tr>
<td>extensionId</td>
<td>required</td>
<td>The extension where the widget is displayed (How does one configure the section to appear in multiple extension points)</td>
</tr>
</tbody>
</table>

Latest Obs For Concept List

Shows a list of concepts and the latest observations for each

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>values</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts</td>
<td>required</td>
<td></td>
<td>Comma delimited list of Concept numeric id, uuid, or mapping</td>
<td>* Provides the concepts whose most recent obs or group observations are returned are returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* Minimum of 1 and a maximum of 10 concepts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* If the concept has no value it is returned with no value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* Text showing how recent the obs is, today, yesterday, n days/months/years ago in smaller italic text whose title is the actual obs date</td>
</tr>
<tr>
<td>maxAge</td>
<td>optional</td>
<td></td>
<td>Numeric integer with a suffix for example: *1d – 1 day *2w – 2 weeks *4m – 4 months from the current date</td>
<td>The maximum age of most recent obs for each of the items in the concept list</td>
</tr>
<tr>
<td>obsGroupLabels</td>
<td>optional</td>
<td>none</td>
<td>*FSN *shortName *none</td>
<td>May be used when obs grouping concept uuid(s) are used in the concept key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies the format in which to display obs group member prefixes, FSN, shortName, none. The prefixes to display refer to either Fully Specified Name, Short Name, of the underlying member concept or No prefix respectively.</td>
</tr>
</tbody>
</table>
Example Latest Obs For Concept List widget app configuration

```json
{
  "id": "coreapps.latestObsForConceptList",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "coreapps.latestObsForConceptList.app.description",
  "order": 10,
  "config": {
    "widget": "latestobsforconceptlist",
    "obsGroupLabels": "FSN",
    "icon": "icon-user-md",
    "label": "Concept List",
    "maxAge": "10d",
    "concepts": "5089AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, 5090AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, 5088AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, 5085AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, 5086AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, 5080AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
  },
  "extensions": [
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardFirstColumn",
      "appId": "coreapps.latestObsForConceptList",
      "extensionPointId": "patientDashboard.firstColumnFragments",
      "extensionParams": {
        "provider": "coreapps",
        "fragment": "dashboardwidgets/dashboardWidget"
      }
    }
  ]
}
```

Image Sample without group obs concept

![Concept List](image)

**Concept List**

**Weight:** 50kg  
**Last Visit Date:** 2017-01-30  
**CD4 count:** 600

Image Sample of grouped obs

![Special Condition](image)

**Special Condition**

**Specific Condition Group:**  
- Hunger strike  
- Violence related injury  
- Alleged ill treatment

Obs Across Encounters

Shows obs with given concepts across multiple encounters based on date. When a concept has no value in an encounter then the value is left blank. If an encounter has multiple obs with the given concept then they will be displayed in separate rows.

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts</td>
<td>required</td>
<td></td>
<td>Comma delimited list of Concept numeric id, uuid, or mapping of the concepts whose observations are to be shown</td>
</tr>
</tbody>
</table>
### Example Obs Across Encounters widget app configuration

```json
{
  "id": "coreapps.obsAcrossEncounters",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "coreapps.obsAcrossEncounters.app.description",
  "order": 10,
  "config": {
    "widget": "obsacrossencounters",
    "icon": "icon-user-md",
    "label": "Health Trend Summary",
    "concepts": "5087AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,5088AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,5089AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
    "encounterType": 123AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,
    "detailsUrl": "dispensing/patient.page?patientId={{patient.uuid}}",
    "maxRecords": "3",
    "maxAge": "1w",
  },
  "extensions": [
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardFirstColumn",
      "appId": "coreapps.obsAcrossEncounters",
      "extensionPointId": "patientDashboard.firstColumnFragments",
      "extensionParams": {
        "provider": "coreapps",
        "fragment": "dashboardwidgets/dashboardWidget"
      }
    }
  ]
}
```

### Image Sample

**Health Trend Summary**

<table>
<thead>
<tr>
<th>Encounter</th>
<th>Weight</th>
<th>CD4</th>
<th>Viral Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-04-01</td>
<td>50 kg</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>2016-06-09</td>
<td>54 kg</td>
<td>300</td>
<td>800</td>
</tr>
<tr>
<td>2016-09-29</td>
<td>59 kg</td>
<td>100</td>
<td>350</td>
</tr>
</tbody>
</table>
**Obs Graph**

Shows a graph of obs values for a concept with numeric values

**Configuration Parameters**

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conceptId</td>
<td>required</td>
<td></td>
<td>Comma delimited string of concepts UUIDs whose values are to be graphed. Only concepts with a numeric data type are to be graphed</td>
</tr>
<tr>
<td>maxResults</td>
<td>required</td>
<td>4</td>
<td>The maximum number of observations to be graphed</td>
</tr>
</tbody>
</table>
| maxAge          | optional |          | The maximum age of most recent obs for each of the items in the concept list  
|                 |          |          | Numeric integer with a suffix for example:  
|                 |          |          | *1d – 1 day  
|                 |          |          | *2w – 2 weeks  
|                 |          |          | *4m – 4 months from the current date                                                  |
| encounterTypes  | optional |          | Comma delimited string of Encounter Type UUIDs. Only the observations from the given list of encounter types will be taken for drawing graph. |
Example Obs Graph widget app configuration

```json
{
    "id": "coreapps.obsGraph",
    "instanceOf": "coreapps.template.dashboardWidget",
    "description": "coreapps.obsGraph.app.description",
    "order": 10,
    "config": {
        "widget": "obsgraph",
        "icon": "icon-bar-chart",
        "label": "Obs Graph",
        "conceptId": "5090AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,3ce93cf2-26fe-102b-80cb-0017a47871b2",
        "maxResults": "6",
        "maxAge": "1w",
        "encounterTypes": "f120a111-dd2a-4a03-b184-c36b1969c827,e63b5ad2-4470-4d22-8c39-661b534043de"
    },
    "extensions": [{
        "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardFirstColumn",
        "appId": "coreapps.obsGraph",
        "extensionPointId": "patientDashboard.firstColumnFragments",
        "extensionParams": {
            "provider": "coreapps",
            "fragment": "dashboardwidgets/dashboardWidget"
        }
    }]
}
```

Image Sample
Data Integrity Violations

Shows the data integrity module violations for a patient

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxResults</td>
<td>required</td>
<td>6</td>
<td>The maximum number of violations to be displayed. A link to fix the violation is only available if the action_url column has a value</td>
</tr>
</tbody>
</table>
Example Data Integrity widget app configuration

```json
{
  "id": "coreapps.dataIntegrityViolations",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "coreapps.dataIntegrityViolations.app.description",
  "order": 10,
  "config": {
    "widget": "dataintegrityviolations",
    "icon": "icon-question-sign",
    "label": "Data Quality Violations",
    "maxResults": "8"
  },
  "extensions": [ 
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardFirstColumn",
      "appId": "coreapps.dataIntegrityViolations",
      "extensionPointId": "patientDashboard.firstColumnFragments",
      "extensionParams": { 
        "provider": "coreapps",
        "fragment": "dashboardwidgets/dashboardWidget"
      }
    }
  ]
}
```

Image Sample

### Data Quality Violations

**No ART Start Date**

**Missing TB regimen start**

**Missing TB confirmatory test**

### Relationships

Shows the relationships for a patient

### Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxRecords</td>
<td>required</td>
<td>6</td>
<td>The maximum number of relationships to display</td>
</tr>
<tr>
<td>baseAppPath</td>
<td>optional</td>
<td>/coreapps</td>
<td>Provides the ability to embed the relationships widget in a different module and page than the default coreapps clinician facing dashboard</td>
</tr>
<tr>
<td>dashboardPage</td>
<td>optional</td>
<td>/coreapps/clinicianfacing /patient.page?patientId=patientUuid</td>
<td>Indicates to which page to navigate when clicking on a relative</td>
</tr>
<tr>
<td>providerPage</td>
<td>optional</td>
<td>null</td>
<td>If configured, it indicates the page to navigate if the relationship points to a provider</td>
</tr>
<tr>
<td>includeRelationshipTypes</td>
<td>optional</td>
<td></td>
<td>Comma delimited list of relationship type UUIDs. It provides the ability to filter which relationship types should be allowed to be created. By default all relationship types are displayed</td>
</tr>
<tr>
<td>editPrivilege</td>
<td>optional</td>
<td></td>
<td>Indicates the privilege required to edit patient relationships. If you want to edit relationships via the widget you need to assign &quot;Task: coreapps.editRelationships&quot; privilege to the logged on user.</td>
</tr>
</tbody>
</table>
Example Relationships widget app configuration

```javascript
{
    "id": "coreapps.relationships",
    "instanceOf": "coreapps.template.dashboardWidget",
    "description": "coreapps.relationships.app.description",
    "order": 10,
    "config": {
        "widget": "relationships",
        "icon": "icon-group",
        "label": "Family",
        "maxRecords": "3",
        "basePath": "~/coreapps",
        "editPrivilege": "Task: coreapps.editRelationships",
        "dashboardPage": "~/coreapps/clinicianfacing/patient.page?patientId=patientUuid",
        "providerPage": "~/coreapps/providermanagement/editProvider.page?personUuid={personUuid}",
        "includeRelationshipTypes": "UUID_OF_RELATIONSHIP_TYPE,"
    },
    "extensions": [
        {
            "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardSecondColumn",
            "appId": "coreapps.relationships",
            "extensionPointId": "patientDashboard.secondColumnFragments",
            "extensionParams": {
                "provider": "coreapps",
                "fragment": "dashboardwidgets/dashboardWidget"
            }
        }
    ]
}
```

Image Sample

![Family]

John Anthony Doe (1972-01-02)   Father
Mary Jane Smith (2016-08-06)   Child
Thomas Edison (2011-08-06)    Sibling

Recent Visit by Encounter Type

Shows the recent visits for a patient by encounter type

**Configuration Parameters**

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxRecords</td>
<td>required</td>
<td>6</td>
<td>The maximum number of recent visits to show</td>
</tr>
<tr>
<td>maxAge</td>
<td>optional</td>
<td></td>
<td>The maximum age of most recent obs for each of the items in the concept list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Numeric integer with a suffix for example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*1d – 1 day</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*2w – 2 weeks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*4m – 4 months from the current date</td>
</tr>
<tr>
<td>combineEncounterTypes</td>
<td>optional</td>
<td>true</td>
<td>When true shows a comma delimited value of encounter types that occur on the same visit, when false shows each encounter type on its own row for the same visit. What are the thoughts on this?</td>
</tr>
</tbody>
</table>
Example Visits By Encounter Type widget app configuration

```json
{
  "id": "coreapps.visitByEncounterType",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "coreapps.visitByEncounterType.app.description",
  "order": 10,
  "config": {
    "widget": "visitbyencountertype",
    "icon": "icon-group",
    "label": "Recent Visits",
    "maxRecords": "3",
    "maxAge": "3m",
    "combineEncounterTypes": "true"
  },
  "extensions": [
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardSecondColumn",
      "appId": "coreapps.visitByEncounterType",
      "extensionPointId": "patientDashboard.secondColumnFragments",
      "extensionParams": {
        "provider": "coreapps",
        "fragment": "dashboardwidgets/dashboardWidget"
      }
    }
  ]
}
```

Image Sample

### Recent Visits

<table>
<thead>
<tr>
<th>Date</th>
<th>Program(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-01-15</td>
<td>ANC</td>
</tr>
<tr>
<td>2017-01-15</td>
<td>Laboratory</td>
</tr>
<tr>
<td>2017-02-20</td>
<td>Maternity</td>
</tr>
</tbody>
</table>

### Recent Visits

<table>
<thead>
<tr>
<th>Date</th>
<th>Program(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-01-15</td>
<td>ANC, Laboratory</td>
</tr>
<tr>
<td>2017-02-20</td>
<td>Maternity</td>
</tr>
</tbody>
</table>

Programs List

Shows the programs the patient is currently enrolled in, as well as historical programs. Serves as an access point into programs-specific dashboards.

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dateFor mat</td>
<td>optional</td>
<td>yyyy-MM-dd</td>
<td>Date format to use when displaying dates</td>
</tr>
<tr>
<td>support edPrograms</td>
<td>optional</td>
<td>null</td>
<td>A comma-separated list of program uuids (i.e. uuid of a program in OpenMRS Program table). If specified, will only display patient enrollments in the selected programs. If left empty (default) will list all patient program enrollments, regardless of program.</td>
</tr>
</tbody>
</table>
| enableProgramDashboards | optional | false | If enabled: 1) adds a hyperlink to each patient program enrollment that links to the URL specified by "dashboardPage", and 2) displays a dropdown listing all programs that the patient is not currently enrolled in (limited by supportedPrograms if specified); selecting a program from the list also redirects the user to the page specified by "dashboardPage"
|---|---|---|
| dashboardPage | optional | | Link to follow when a user clicks on a program hyperlink or selected a program to enroll a patient in. "PatientUuid" will be replaced with the uuid, and the "dashboard" will be substituted with the uuid of the selected program, enabling use of Program-Specific dashboards (see Context-Specific Dashboards).
| maxRecords | optional | 20 | Max number of patient program enrollments to display

### Privileges

<table>
<thead>
<tr>
<th>privilege</th>
<th>description</th>
</tr>
</thead>
</table>
| Task: enrollInProgram | Determines whether or not to render dropdown (see example below) allowing users to navigate to a program-specific dashboard to enroll in a program if not enrolled.

### Example Programs List widget app configuration

```json
{
  "id": "coreapps.programs",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "List of Programs",
  "order": 10,
  "config": {
    "widget": "programs",
    "icon": "icon-stethoscope",
    "label": "Programs",
    "dateFormat": "dd MMM yyyy",
    "supportedPrograms": "7b6a71b1-742d-4d39-ac53-8ae6d4db960c,e7f774a5-0929-4c14-bb5c-563c6b390811",
    "enableProgramDashboards": true
  },
  "extensions": [
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardSecondColumn",
      "appId": "coreapps.programs",
      "extensionPointId": "patientDashboard.secondColumnFragments",
      "extensionParams": {
        "provider": "coreapps",
        "fragment": "dashboardwidgets/dashboardWidget"
      }
    }
  ]
}
```

### Image Sample

Example of patient enrolled in two programs:

#### PATIENT PROGRAMS

<table>
<thead>
<tr>
<th>Program</th>
<th>Start Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zika</td>
<td>Aug 1, 2017 - Current</td>
</tr>
<tr>
<td>HIV</td>
<td>May 16, 2017 - Current</td>
</tr>
</tbody>
</table>

Example of dropdown box to navigate to program-specific dashboard to enroll a patient in a program:
Program Status

Shows the specifics of the current patient program enrollment: date enrolled, enrollment location, date completed, and outcome. Allows editing of these four data points, and the deletion of the patient program (assuming the user has the appropriate privilege). If the patient is not currently enrolled in the program, provides the ability to enroll in the program.

An updated version of this widget (likely to be released in Coreapps 1.16.0) provided the ability to set and edit workflow state transitions.

This widget is also used by the Program History widget to allow viewing and editing of completed programs.

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>required</td>
<td></td>
<td>Program UUID (i.e. uuid of a program in OpenMRS Program table)</td>
</tr>
<tr>
<td>dateFormat</td>
<td>optional</td>
<td>yyyy-mm-dd</td>
<td>Date format to use when displaying dates</td>
</tr>
<tr>
<td>locationTag</td>
<td>optional</td>
<td>null</td>
<td>UUID of a location tag. If specified, the options in the &quot;program enrollment location&quot; drop-down will be limited to locations with tag.</td>
</tr>
</tbody>
</table>

Privileges

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task: enrollInProgram</td>
<td>Whether the user has the ability to enroll a patient in a program</td>
</tr>
<tr>
<td>Task: editPatientProgram</td>
<td>Whether the user has the ability to edit an existing patient program</td>
</tr>
<tr>
<td>Task: deletePatientProgram</td>
<td>Whether the user has the ability to delete an existing patient program</td>
</tr>
</tbody>
</table>
Example Visits By Encounter Type widget app configuration

```
{
  "id": "coreapps.programStatus",
  "instanceOf": "coreapps.template.dashboardWidget",
  "description": "List of Programs",
  "order": 10,
  "config": {
    "widget": "programstatus",
    "icon": "icon-stethoscope",
    "label": "Program Status",
    "dateFormat": "dd MMM yyyy",
    "program": "7b6a71b1-742d-4d39-ac53-8ae6d4db960c",
    "locationTag": "ef54a24a-dd76-4636-b94d-f7b486107369"
  },
  "extensions": [{
    "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardSecondColumn",
    "appId": "coreapps.programStatus",
    "extensionPointId": "7b6a71b1-742d-4d39-ac53-8ae6d4db960c.firstColumnFragments",
    "extensionParams": {
      "provider": "coreapps",
      "fragment": "dashboardwidgets/dashboardWidget"
    }
  }]
}
```

Image Samples

Prior to enrollment:

![CURRENT ENROLLMENT](image1.png)

After enrollment:

![CURRENT ENROLLMENT](image2.png)

Editing enrollment:

![CURRENT ENROLLMENT](image3.png)
Program History

Given a program, shows all of a patient's enrollments in that program by rendering a Program Status widget (see above) for each program, ordered from most recent to oldest.

Configuration Parameters

<table>
<thead>
<tr>
<th>key</th>
<th>required</th>
<th>default</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>program</td>
<td>required</td>
<td></td>
<td>Program UUID (ie. uuid of a program in OpenMRS Program table)</td>
</tr>
<tr>
<td>dateFormat</td>
<td>optional</td>
<td>yyyy-MM-dd</td>
<td>Date format to use when displaying dates</td>
</tr>
<tr>
<td>locationTag</td>
<td>optional</td>
<td>null</td>
<td>UUID of a location tag. If specified, the options in the &quot;program enrollment location&quot; drop-down will be limited to locations with tag.</td>
</tr>
<tr>
<td>includeActive</td>
<td>optional</td>
<td>true</td>
<td>If set to false, excludes the active program enrollment (if any) from the list</td>
</tr>
</tbody>
</table>

Example Visits By Encounter Type widget app configuration

```json
{
  "id": "coreapps.programHistory",
  "order": 10,
  "config": {
    "icon": "icon-stethoscope",
    "label": "Previous Enrollment",
    "dateFormat": "dd MMM yyyy",
    "program": "7b6a71b1-742d-4d39-ac53-8ae6d4db960c",
    "locationTag": "ef54a24a-dd76-4636-b94d-f7b486107369",
    "includeActive": "false"
  },
  "extensions": [
    {
      "id": "org.openmrs.module.coreapps.mostRecentVitals.clinicianDashboardSecondColumn",
      "appId": "coreapps.programHistory",
      "extensionPointId": "7b6a71b1-742d-4d39-ac53-8ae6d4db960c.secondColumnFragments",
      "extensionParams": {
        "provider": "coreapps",
        "fragment": "program/programHistory"
      }
    }
  ]
}
```
Bahmni Appointments Widget (Capability).

This widget reports on Bahmni Appointments module. It is disabled by default because the Bahmni Appointments module is not part of the Reference Application.

To have this widget show up on patientDashboard when running Bahmni Appointments module, the implementer will have to add the following configuration in a json file named as <name>_app.json under /omod/src/main/resources/app/:
Bahmni Appointments widget app configuration example

```
{
    "id": "coreapps.Appointments",
    "instanceOf": "coreapps.template.dashboardWidget",
    "description": "coreapps.appointments.app.description",
    "order": 14,
    "config": {
        "widget": "bahmniappointments",
        "icon": "icon-calendar",
        "label": "APPOINTMENTS",
        "editPrivilege": "App: appointmentschedulingui.viewAppointments",
        "maxRecords": "5",
        "editIcon": "icon-share-alt",
        "detailsUrl": "appointments/appointments"
    },
    "extensions": [
        {
            "id": "${project.parent.groupId}.${project.parent.artifactId}.Appointments.clinicianDashboardSecondColumn",
            "appId": "coreapps.Appointments",
            "extensionPointId": "patientDashboard.secondColumnFragments",
            "extensionParams": {
                "provider": "${project.parent.artifactId}",
                "fragment": "dashboardwidgets/dashboardWidget"
            }
        }
    ]
}
```

NB. The following params should be left constant as below ie

- "instanceOf": "coreapps.template.dashboardWidget"
- config.widget: "bahmniappointments"
- extensions.extensionParams.provider: "${project.parent.artifactId}",
- extensions.extensionParams.fragment: "dashboardwidgets/dashboardWidget",

Note. The param "detailsUrl": "url", is the url to redirect the user to the appointments module.

**Image Sample**

![Appointments Image](image-url)

There are a number of additional features that would be helpful on the dashboard: Future enhancements

- obsgroup table
  - Similar to the obsacrossencounters but for obsgroups. Should allow for multiple obs within an obsgroup.
  - Use case: A list of medications dispensed with encounter date, medication name, and frequency
- Graphs
  - Graphs with log scale
  - Use case: Viral load scale could vary from non-detectable to 10,000. It would be helpful to graph.
  - Multiple variable graphs
  - Use case: Graph 2 or more variables (concepts) on the same graph. Use for weight and height.
• BMI graph
  • Use case: Based on weight and height, graph weight and calculated BMI on a single graph.